Designing with VHDL
FPGA 1

Course Specification

- Composite Data Types (Lecture)
- VHDL Operators (Lecture)
- Concurrency in VHDL (Lecture)
- Concurrent Assignments (Lecture, Lab)
- Processes and Variables (Lecture, Demo, Lab)

Day 2

- Conditional Statements in VHDL: if/else, case (Lecture, Lab)
- Sequential Looping Statements (Lecture, Lab)
- Delays in VHDL: Wait Statements (Lecture)
- Introduction to the VHDL Testbench (Lecture, Lab)
- VHDL Assert Statements (Lecture)
- VHDL Attributes (Lecture)
- VHDL Subprograms (Lecture)
- VHDL Functions (Lecture, Lab)
- VHDL Procedures (Lecture)

Day 3

- VHDL Libraries and Packages (Lecture, Lab)
- Interacting with the Simulation (Lecture)
- Finite State Machine Overview (Lecture)
- Mealy Finite State Machine (Lecture)
- Moore Finite State Machine (Lecture, Lab)
- FSM Coding Guidelines (Lecture)
- Vivado Simulator and Race Conditions in VHDL (Lecture)
- Writing a Good Testbench (Lecture, Lab)
- Targeting Xilinx FPGAs (Lecture, Lab)

Topic Descriptions

Day 1

- Introduction to VHDL – Discusses the history of the VHDL language and provides an overview of the different features of VHDL.
- VHDL Design Units – Provides an overview of typical VHDL code.
- VHDL Objects, Keywords, Identifiers – Discusses the data objects that are available in the VHDL language as well as keywords and identifiers.
- Scalar Data Types – Covers both intrinsic and commonly used data types.
- Composite Data Types – Covers composite data types (arrays and records).
- VHDL Operators – Reviews all VHDL operator types.
- Concurrency in VHDL – Describes concurrent statements and how signals help in achieving concurrency.
- Concurrent Assignments – Covers both conditional and unconditional assignments.
- Processes and Variables – Introduces sequential programming techniques for a concurrent language. Variables are also discussed.

Day 2

- Conditional Statements in VHDL: if/else, case – Describes conditional statements such as if/else and case statements.
- Sequential Looping Statements – Introduces the concept of looping in both the simulation and synthesis environments.
- Delays in VHDL: Wait Statements – Covers the wait statement and how it controls the execution of the process statement.
- Introduction to the VHDL Testbench – Introduces the concept of the VHDL testbench.

Course Outline

Day 1

- Introduction to VHDL (Lecture)
- VHDL Design Units (Lecture, Lab)
- VHDL Objects, Keywords, Identifiers (Lecture)
- Scalar Data Types (Lecture)

Day 2

- VHDL Operators (Lecture)
- Concurrency in VHDL (Lecture)
- Concurrent Assignments (Lecture, Lab)
- Processes and Variables (Lecture, Demo, Lab)

Day 3

- VHDL Libraries and Packages (Lecture, Lab)
- Interacting with the Simulation (Lecture)
- Finite State Machine Overview (Lecture)
- Mealy Finite State Machine (Lecture)
- Moore Finite State Machine (Lecture, Lab)
- FSM Coding Guidelines (Lecture)
- Vivado Simulator and Race Conditions in VHDL (Lecture)
- Writing a Good Testbench (Lecture, Lab)
- Targeting Xilinx FPGAs (Lecture, Lab)
Designing with VHDL
FPGA 1

Course Specification

- VHDL Assert Statements – Describes the concept of VHDL assertions.
- VHDL Attributes – Describes attributes, both predefined and user defined.
- VHDL Subprograms – Covers the use of subprograms in verification and RTL code to model functional blocks.
- VHDL Functions – Describes functions, which are integral to reusable and maintainable code.
- VHDL Procedures – Describes procedures, common constructs that are also important for reusing and maintaining code.

Day 3
- VHDL Libraries and Packages – Demonstrates how libraries and packages are declared and used.
- Interacting with the Simulation – Describes how to interact with a simulation via text I/O.
- Finite State Machine Overview – Provides an overview of finite state machines, one of the more commonly used circuits.
- Mealy Finite State Machine – Describes the Mealy FSM and how to code for it.
- Moore Finite State Machine – Describes the Moore FSM and how to code for it.
- FSM Coding Guidelines – Discusses FSM implementation in an FPGA using VHDL.
- Vivado Simulator and Race Conditions in VHDL – Introduces the Vivado simulator simulation environment. Race conditions are also discussed.
- Writing a Good Testbench – Explores how time-agnostic, self-checking testbenches can be written and applied.
- Targeting Xilinx FPGAs – Focuses on Xilinx-specific implementation and chip-level optimization.

Register Today
Xilinx’s network of Authorized Training Providers (ATP) delivers public and private courses in locations throughout the world. Please contact your closest ATP for more information, to view schedules, or to register online. Visit www.xilinx.com/training and click on the region where you want to attend a course.

Americas, contact your training provider at www.xilinx.com/training/atp.htm#NA or send your inquiries to registrar@xilinx.com.

Europe, contact your training provider at www.xilinx.com/training/atp.htm#EU or send your inquiries to eurotraining@xilinx.com.

Asia Pacific, contact your training provider at www.xilinx.com/training/atp.htm#AP, or send your inquiries to education_ap@xilinx.com, or call +852-2424-5200.

Japan, contact your training provider at www.xilinx.com/training/atp.htm#JP, or send your inquiries to education_kk@xilinx.com, or call +81-3-6744-7970